Concept Curve Paradigm

A new approach to Knowledge representation in the AI era

Daniel Bistman – daniel.bistman@gmail.com

Bachelor of Business Administration - Independent AI researcher.

Abstract

Current knowledge representation techniques in Artificial Intelligence, particularly high dimensional embeddings, face significant limitations when handling complex, structured information like extensive narratives or large bodies of knowledge.

Representing rich semantic structures as single vectors often leads to information compression, loss of meaning, and potential hallucinations in generative models. This paper introduces the Concept Curve Paradigm, a novel approach that redefines knowledge representation by modeling concepts, stories, and reasoning sequences not as isolated points, but as dynamic networks or trajectories of interrelated concepts within a semantic space.

This new paradigm preserves the inherent structure and relationships within information, overcoming the constraints of static embeddings. We detail Concept Curve Embeddings Indexation (CC-EI), a practical method derived from this paradigm, which indexes information fragments based on their key conceptual interconnections rather than compressing them into dense vectors.

The Concept Curve approach offers numerous benefits, including eliminating redundancy, enabling flexible conceptual connections, enhancing AI reasoning, facilitating unlimited context input and output, improving computational efficiency, and potentially shifting AI bottlenecks away from compute constraints.

Overall, the Concept Curve Paradigm offers a new foundation for more scalable, interpretable, and capable AI systems.

All methods described in this paper are publicly implemented and freely available through open source code and documentation.

Introduction

The rapid evolution of Artificial Intelligence, particularly in the domain of Large Language Models, has underscored the critical role of knowledge representation. While high dimensional embeddings have driven significant progress, their capacity to accurately capture complex, structured information and extended narratives remains a fundamental challenge, often leading to semantic loss and hindering model scalability and interpretability.

This paper confronts these limitations by introducing the Concept Curve Paradigm, a novel framework that conceptualizes knowledge not as static points in vector space, but as dynamic trajectories of interconnected concepts.

Structure of This Paper

The paper will be divided into two sections:

In the First Section

1. Embeddings: A Journey from Their Origins to Their Limits - we explore Embeddings from their origins to the present day and their limitations.
2. The Birth of the Concept Curve Paradigm - we introduce the proposed solution to current state of the art limitations.
3. The Concept Curve Embeddings Indexation - we explain a new model-agnostic indexing method for the future of AI, the Concept Curve Embeddings Indexation (CC-EI).
4. Conclusion
5. Note on Benchmarks
6. References

In the Second Section we present a series of explanatory appendices on the practical use of the Concept Curve paradigm (Examples and Details).

Annex 1 – Unlimited Size Input Context

Annex 2 – Computational Savings in Query Processing

Annex 3 – Unlimited Size Output

Annex 4 – Computational Savings in Output Processing

Annex 5 – No Longer Compute Constrained

Annex 6 – A Solution to Visual Stickiness in AI Image Outputs.

Annex 7 – Advanced Image Recognition and Semantic Explanation

Annex 8 – Real Time Knowledge Updating

While the first section focuses on introducing the paradigm and its methodological foundations in a conceptual and illustrative manner, the second section (Annexes) includes a series of applied examples and technical formulations. These annexes provide formal expressions and performance-related considerations that support the theoretical framework presented above.

First Section

1. Embeddings: A Journey from Their Origins to Their Limits

1.1 - What Are Embeddings?

In the context of Natural Language Processing (NLP), embeddings are dense numerical representations of words, phrases, or tokens in the form of vectors in a high dimensional space.

These representations capture semantic and syntactic relationships so that words with similar meanings are located close to one another in that vector space.

1.2 - What Are They Used For?

Embeddings enable machines to understand and process human language mathematically. They serve as a foundation for tasks such as text classification, machine translation, sentiment analysis, question answering, and text generation.

Thanks to embeddings, models can distinguish between different uses of the same word (e.g., “bank” as a bench vs. “bank” as a financial institution) and reason about meanings, analogies, and context with remarkable precision.

1.3 - The Birth of Modern Embeddings

Before the term ‘embeddings’ was formally adopted, earlier efforts such as the Neural Probabilistic Language Model (Bengio et al., 2003) [1] laid theoretical foundations for distributed representations of language.

The true turning point came with the 2013 paper by Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean titled “Efficient Estimation of Word Representations in Vector Space” [2].

This work laid the groundwork for what we now call embeddings, enabling models to capture semantic relationships with impressive effectiveness.

A Google search could now disambiguate “apple” as either a fruit or a technological Global Company, based on context.

1.4 - What Are Dimensions? How Many Dimensions Do Modern Models Have?

The initial Word2Vec models trained by Google used various vector sizes, but the publicly released model had 300 dimensions [3] with a vocabulary of approximately 3 million words and phrases (tokenized as compound tokens, akin to n-grams).

Fast forward in time, current models differ significantly from Google's 2013–2016 design: modern LLMs like GPT use vocabularies of about 100,000 subword tokens instead of 3 million n-grams, and they employ over 12,000 dimensions per token rather than the original 300 (e.g., GPT-3 "Davinci" uses 12,288 dimensions).

1.5 - Interim Observations

Having understood what embeddings are in modern models, we can restate the concept in other words:

> *"An embedding is the vector representation of a concept, expressed as a point in a high dimensional space."*

For example, to capture the meaning of the word "***bird***", the model translates it into a vector, a specific point in a mathematical space of over 12,000 dimensions.

If we analyze a sentence like "***the bird flies across the blue sky***" each token ("bird", "flies", "sky", "blue") is also represented as a vector in that same space, with its meaning adjusted according to context.

Thus, embeddings allow us not only to encode individual words but also to model complex contextual relationships, preserving subtle meaning variations that shift dynamically with the sentence.

1.6 - The Limitations of Embeddings

Initially, embeddings were used to represent single words ("***city***")… then they expanded to represent compound concepts ("***new_york_city***")… gradually, they were applied to phrases, then **paragraphs**… and even entire **documents**…

…This escalation exposed a clear technical boundary. The limit became apparent when trying to represent full books (for example, ***Gulliver's Travels***) with a single vector. This revealed the technique's inadequacy.

> *Representing a word like "**bird**" as a point in a 12,000 dimensional space is possible, perhaps even redundant. But capturing the full semantic richness and narrative of **Gulliver's Travels** in that same space is clearly insufficient.*

Since around 2020, studies such as Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks - Lewis et al., 2020 [4] have confirmed that an embedding alone cannot encapsulate the complexity of structured knowledge, a complete story, or a broad conceptual framework. In these cases, the information compression forced by embeddings leads to semantic loss, ambiguity, and— in generative systems—**hallucinations**.

1.7 – Preliminary conclusion

If the core limitations of current large language models arise not from lack of scale, but from the underlying architecture of semantic representation, then a new paradigm is required, one that does not attempt to compress meaning into fixed vectors, but instead embraces the fluidity, temporal depth and emergent structure of concepts. This is how the Concept Curve Paradigm emerged.

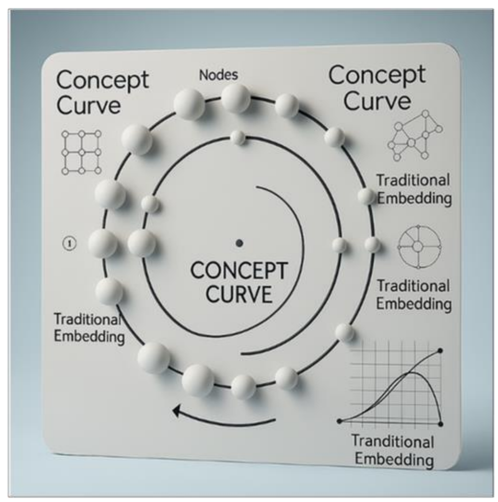
2. The Birth of the Concept Curve Paradigm

2.1 Definition

The Concept Curve paradigm establishes that Knowledge, Stories, or Reasoning Sequences should not be represented as a single point in a multidimensional space, but rather as a network of interrelated simpler concepts.

2.2 Conceptual Representation: From Points to Trajectories

Rather than compressing meaning into a single point, this approach models conceptual trajectories through an ordered semantic space. By representing sequences of concepts as curves, information remains structured and traceable, enabling the representation of complex stories, ideas, or bodies of knowledge without sacrificing semantic fidelity.



In current **Transformer based architectures** (Vaswani et al., 2017) [5] such as GPT, BERT, or T5, concepts like “principle of gravity”, if embedded, will typically be embedded as dense vectors in spaces with thousands of dimensions, often ranging from 768 to over 12,000 dimensions, depending on model size and architecture:

- GPT-3 uses 12,288 dimensions in its largest variant.
- BERT-Large uses 1024.
- T5-11B uses similar high dimensional spaces.

Under embedding, a concept like “principle of gravity”, if stored, would be done with over 12 thousand float values:

principle_of_gravity = [0.182…, -0.537…, 0.901…, -0.244…, -0.510…, 0.921…, 0.333…, 0.627…, -0.389…, 0.577… … …]

Storing a single 12,288 dimensional float32 embedding costs approx. ~48 KB.

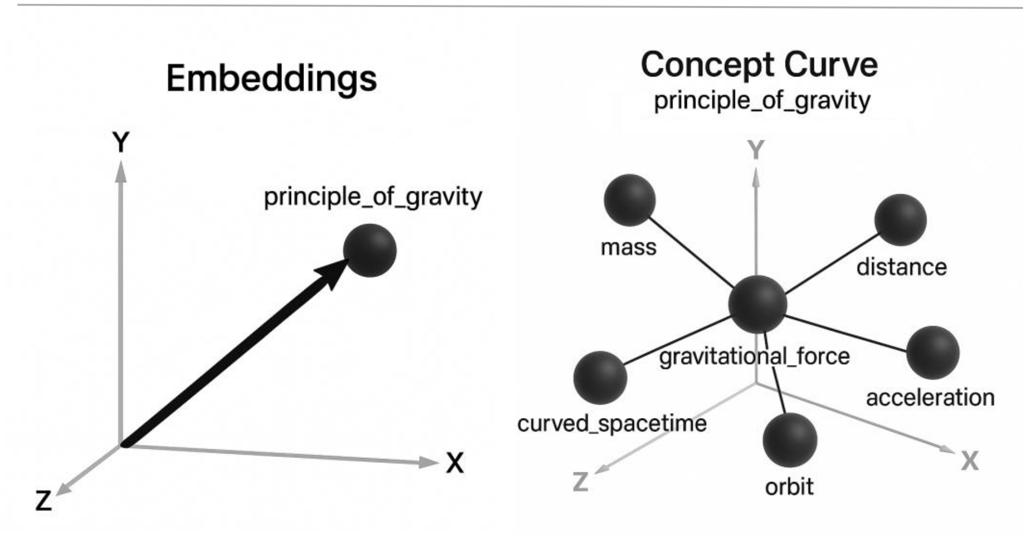
Instead, under the **Concept Curve paradigm**, the concept “principle of gravity” can be decomposed into the following six simpler, interrelated conceptual nodes:

Mass — the property that gives rise to gravitational attraction.
Distance — the spatial separation influencing gravitational strength.
Gravitational Force — the attractive interaction between masses.
Acceleration — the motion response caused by gravity.
Orbit — the stable motion resulting from gravitational balance.
Curved Spacetime — the geometric interpretation of gravity in relativity

And be now stored like this, six concepts represented in words:

principle_of_gravity = [mass, distance, gravitational_force, acceleration, orbit, curved_spacetime]

a storage cost of approx. ~0.06 kb



Storage ≈ 48 kb

Storage ≈ 0.06 kb

Yes, in the old embedding paradigm, “principle_of_gravity” was compressed into a single high dimensional vector, where semantic, syntactic, and contextual properties are entangled. This encoding lacks structured transparency and does not reflect internal concept composition explicitly.

But in this new CC paradigm, principle_of_gravity is now a concept represented as “a cloud of concepts”.

2.3 The Nature of Meaning in Concept Curves

While traditional embeddings obscure internal structure by entangling semantics in opaque vectors, Concept Curves make the compositional semantics of a concept explicitly accessible.

This aligns with the demand for structural interpretability (Lipton, 2018) [6], in which not only the outputs but also the internal reasoning paths are understandable to humans.

The human brain, while it uses base interpretations that can resemble embeddings (pure abstractions), not all abstractions act the same way.

For complex concepts, the human activates diverse knowledge nodes in the mind, which, interrelated with each other, form that complex knowledge. This is where the Concept Curve paradigm comes in.

Before, when we used embeddings we said “the meaning is around the vector” but now, with this new paradigm, we understand the meaning of something as we say “The meaning is in the cloud”, giving now a richer and complete interpretation with far less storage.

2.4 Foundational Benefits of this paradigm

- Eliminates redundancy in knowledge representation, optimizing storage and processing.

- Distinguishes between structural and sequential knowledge, enabling a more precise construction of meaning.

- Promotes flexible conceptual interconnection, enhancing the AI's ability to reason and generate relevant content.

2.5 Origin of the Name: "Concept Curve"

It is called Concept Curve because it represents thought as a continuous, flexible, and evolving trajectory where concepts dynamically interconnect, not as fixed points or rigid structures.

3. The Concept Curve Embeddings Indexation (CC-EI)

3.1 Definition

The Concept Curve Embeddings Indexation, from now CC-EI, is an indexing method derived from the Concept Curve paradigm. In this approach, texts or information fragments ("chunks") are not compressed into a single vector space via traditional embeddings but are indexed according to their **key concept interconnections**.

3.2 The Method in Practice

The method consists in representing any complex concept, idea, document fragment, complete book, or knowledge structure not as a “vector,” but as a network of conceptual nodes, a cloud of interconnected concepts.

From this point forward, we can represent the concept “industrial revolution” as follows:

> **industrial_revolution** = **[**steam_engine, industrialization, factories, mass_production, urbanization, capitalism, proletariat, trade_unionism, railway, technological innovation, division_of_labor, pollution, agricultural_revolution, textiles, mining, hydropower, working_class, social_inequality, social_mobility, imperialism**]**

K = 20 concepts - Estimated storage ≈ 0.36 KB

And we can represent the classic novel Oliver Twist as follows:

> **Oliver_Twist** = **[**orphanhood, poverty, child_labor, criminality, oppressive_institutions, innocence, virtue_versus_evil, redemption, social_class, fate, city_vs_countryside, Victorian_London, social_hypocrisy, justice, lost_family, identity, child_exploitation, institutional_corruption, compassion, friendship, resilience, morality, sacrifice, crime_network, social_reform, inequality, hope, abuse_of_power, survival, personal_transformation**]**

K = 30 concepts - Estimated storage ≈ 0.44 KB

3.3 Applicability

Using this method, when dealing with documents of 1 million tokens or more, it is no longer necessary to split them into chunks and embed them as dense vectors. Instead, chunks can be indexed using groups of interrelated concepts. This enables speed, efficiency, and robustness.

Documents or chunks will no longer lose interpretability; on the contrary, retrieval becomes easier since the semantic representation is richer than in traditional vector embedding.

The method is implemented and demonstrated in the source code attached to this paper. Repository of the demonstration software: [tinyurl.com/CCEI-gHub] [7] This source code confirms the method's functionality and scalability.

A practical example of indexing the Uniform Commercial Code of the State of Michigan could be represented as follows:

> **File: Chunk_Art01.txt** = **[**general_provisions, UCC_applicability, liberal_construction, uniformity_of_law, implied_repeal, electronic_signatures, general_definitions, aggrieved_party, buyer_in_ordinary_course, document_of_title, security_interest, good_faith_obligation, usage_of_trade, lease_vs_security_interest, presumption, notice_and_knowledge, reasonable_time, subordination, course_of_performance, prima_facie_evidence**]**

K = 30 concepts - Estimated storage ≈ 0.42 KB

In contrast, using a current embedding method with a 12,288 dimensional vector would require ≈ 48 KB of storage, along with computational costs for embedding and later comparison during retrieval.

3.4 The Indexing Process: How is it made?

The indexing is not performed manually. The indexing is **made automatically**, simply ask the AI: "Give me a group of 30 concepts that represent this document".

That's it, is how the CC-EI indexing is made.

3.5 Retrieval Using CC-EI

Once document chunks have been indexed via CC-EI, retrieval is achieved with higher robustness and much lower computational cost than the older embeddings method.

The freely shared software functions using CC-EI method, the process of retrieval can be illustrated through an analogy: it is similar to a student entering a library with a question written on paper (query).

The student does not review all the books in the library to get the answer, but rather, approaching the shelves with the related subject matter, glances at the books' indexes[1], noting down on paper which pages of which books are worth checking.

---

[1] Not an Embedding index but a Concept Curve index.

Then, they proceed to browse the different books, not in their entirety, but only the chunks previously noted as “worthy of examination”.

The student also does not need to examine all the chunks; they will find one, two, three, or more valid answers to their question (query), note down each valid answer in a notebook, and will stop their search when they feel satisfied.

Finally, from all the noted valid answers, they will create a final summary.

The method is extremely fast and economical because it does not compress embeddings, does not calculate comparisons between embeddings, but only looks at a conceptual index, as the student in the analogy would.

Furthermore, for immense documents, it does not need to review the entire content, but rather, by looking at the index, ends up reviewing a fraction of the complete documents.

The method is model-agnostic, as indexing can be performed by one model and retrieval successfully executed by another.

Cross-lingual capability has also been confirmed through multiple tests: indexing performed in Spanish can be successfully retrieved in English by an LLM, and vice versa. This highlights the sophisticated semantic understanding leveraged by modern AIs within this framework[2].

3.6 Origin of the Name: "Embeddings Indexation"

The term 'Embeddings Indexation' originates from its use of a vocabulary aligned with classic embeddings but uses these concepts as readable, connectable semantic units. The embedding ceases to be a number and becomes a verb.

3.7 Advantages Over Traditional Methods

1. Lighter in processing and storage
2. Avoids static information representation
3. Facilitates human interpretation and AI transparency
4. Reduces dimensionality without loss of meaning
5. Optimizes Information Retrieval (IR)
6. Captures knowledge structure more effectively
7. Compatible with any LLM model, online or offline, present or future technology
8. Never obsolete: its structure adapts to AI evolution

[2] The method is verified in the source code attached to this paper. Repository of the demonstration software: tinyurl.com/CCEI-gHub

4. Conclusion

The Concept Curve Paradigm represents a fundamental shift in how knowledge is structured, retrieved, and interpreted within artificial intelligence systems. By replacing high-dimensional vector compression with networks of semantically connected concepts, it introduces a model-agnostic architecture that is both lightweight and structurally transparent.

This paradigm redefines several layers of the AI stack: it reduces computational overhead through concept level representation, extends memory capacity via dynamic semantic indexing, and optimizes both real time retrieval and generative processes.

Its modularity supports the scalable construction of responses without context loss, while enhancing interpretability in both text based and multimodal environments.

The CC-EI method, demonstrated in practice through publicly available open source code, confirms the paradigm's operational viability. Rather than merely theorizing an alternative to embeddings, it delivers a working framework that transforms traditional bottlenecks such as memory limits, retrieval inefficiencies, and opaque reasoning into opportunities for coherence, efficiency, and adaptability.

In this light, Concept Curve is not just a theoretical model, but a foundation for a new generation of intelligent systems: systems that remember precisely, reason modularly, and scale transparently.

5. Note on Benchmarks

Unlike many technical papers that include static benchmark tables or controlled test environments, this work does not rely on pre-packaged performance metrics. Instead, we offer something more fundamental: publicly available source code and complete documentation of its functioning that allows any reader to reproduce and verify the claims using any sufficiently advanced LLM without requiring access to proprietary frameworks or closed infrastructure.

The Concept Curve Paradigm and the CC-EI method are not proposed as abstract theories but as operational tools. The emphasis is on **direct demonstrability**, not on synthetic benchmarks detached from real world adaptability. Any practitioner, developer, or researcher can validate the method’s capabilities by running the code and observing the retrieval behavior and memory dynamics in action.

In this way, reproducibility becomes empirical, not through institutional resources or benchmark citations, but through **open, architecture-agnostic execution**.

6. References

1. Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A Neural Probabilistic Language Model. Journal of Machine Learning Research, 3(Feb), 1137–1155.

   https://www.jmlr.org/papers/volume3/bengio03a/bengio03a.pdf

2. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013) Efficient Estimation of Word Representations in Vector Space.

   https://arxiv.org/abs/1301.3781

3. Google Inc. (2013). Word2Vec: Tool for Computing Continuous Distributed Representations of Words. Google Code Project (archived).

   https://code.google.com/p/word2vec

4. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Riedel, S. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.

   https://arxiv.org/pdf/2005.11401

5. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems, 30 (NeurIPS 2017)

   https://arxiv.org/pdf/1706.03762

6. Lipton, Z. C. (2018). The mythos of model interpretability. Communications of the ACM, 61(10), 36–43.

   https://arxiv.org/pdf/1606.03490

7. Source Code for the Concept Curve method demonstration, information & documentation.

   https://tinyurl.com/CCEI-gHub

   https://github.com/Daniel-codi/Concept_Curve_Embeddings_Indexation

This preliminary version of the First Section of the Concept Curve paper is released publicly on May 11th, 2025